```
SYSTEM:OS  - DIALOG OneSearch
  File  15:ABI/Inform(R)  1971-2003/Feb 06
         (c) 2003 ProQuest Info&Learning
*File  15: Alert feature enhanced for multiple files, duplicate
removal, customized scheduling. See HELP ALERT.
  File  16:Gale Group PROMT(R)  1990-2003/Feb 05
         (c) 2003 The Gale Group
*File  16: Alert feature enhanced for multiple files, duplicate
removal, customized scheduling. See HELP ALERT.
  File 148:Gale Group Trade & Industry DB  1976-2003/Feb 06
         (c)2003 The Gale Group
*File 148: Alert feature enhanced for multiple files, duplicate
removal, customized scheduling. See HELP ALERT.
  File 160:Gale Group PROMT(R)  1972-1989
         (c) 1999 The Gale Group
  File 275:Gale Group Computer DB(TM)  1983-2003/Feb 05
         (c) 2003 The Gale Group
  File 621:Gale Group New Prod.Annou.(R)  1985-2003/Feb 04
         (c) 2003 The Gale Group
  File   9:Business & Industry(R)  Jul/1994-2003/Feb 05
         (c) 2003 Resp. DB Svcs.
  File  20:Dialog Global Reporter  1997-2003/Feb 06
         (c) 2003 The Dialog Corp.
  File 476:Financial Times Fulltext  1982-2003/Feb 06
         (c) 2003 Financial Times Ltd
  File 610:Business Wire  1999-2003/Feb 06
         (c) 2003 Business Wire.
*File 610: File 610 now contains data from 3/99 forward.
Archive data (1986-2/99) is available in File 810.
  File 613:PR Newswire  1999-2003/Feb 06
         (c) 2003 PR Newswire Association Inc
*File 613: File 613 now contains data from 5/99 forward.
Archive data (1987-4/99) is available in File 813.
  File 624:McGraw-Hill Publications  1985-2003/Feb 05
         (c) 2003 McGraw-Hill Co. Inc
  File 634:San Jose Mercury   Jun 1985-2003/Feb 05
         (c) 2003 San Jose Mercury News
  File 636:Gale Group Newsletter DB(TM)  1987-2003/Feb 05
         (c) 2003 The Gale Group
  File 810:Business Wire  1986-1999/Feb 28
         (c) 1999 Business Wire
  File 813:PR Newswire  1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
  File   2:INSPEC  1969-2003/Jan W4
         (c) 2003 Institution of Electrical Engineers
*File   2: Alert feature enhanced for multiple files, duplicates
removal, customized scheduling. See HELP ALERT.
  File  35:Dissertation Abs Online  1861-2003/Jan
         (c) 2003 ProQuest Info&Learning
  File  65:Inside Conferences  1993-2003/Feb W1
         (c) 2003 BLDSC all rts. reserv.
  File  99:Wilson Appl. Sci & Tech Abs  1983-2003/Dec
         (c) 2003 The HW Wilson Co.
  File 233:Internet & Personal Comp. Abs.  1981-2003/Jan
         (c) 2003 Info. Today Inc.
  File 256:SoftBase:Reviews,Companies&Prods.  82-2003/Jan
         (c)2003 Info.Sources Inc
  File 474:New York Times Abs  1969-2003/Feb 04
         (c) 2003 The New York Times
  File 475:Wall Street Journal Abs  1973-2003/Feb 05
         (c) 2003 The New York Times
  File 583:Gale Group Globalbase(TM)  1986-2002/Dec 13
         (c) 2002 The Gale Group
*File 583: This file is no longer updating as of 12-13-2002.
```

```
Ref    Items  Index-term
E1         0 *AU=ERNEST, L
E2         1  AU=ERNEST, L. R.
E3         1  AU=ERNEST, L.A.
E4         1  AU=ERNEST, LAWRENCE A
E5         1  AU=ERNEST, LELAND CARLYLE
E6         5  AU=ERNEST, LESLIE
E7         4  AU=ERNEST, M.
E8         1  AU=ERNEST, M.J.
E9         2  AU=ERNEST, M.W.
E10        1  AU=ERNEST, MICHAEL JEFFREY
E11        1  AU=ERNEST, MICHAEL JULES
E12        1  AU=ERNEST, MICHAEL V

          Enter P or PAGE for more
?p

Ref    Items  Index-term
E13        2  AU=ERNEST, MICHAEL V JR
E14        1  AU=ERNEST, MICHAEL VANCE, JR.
E15       18  AU=ERNEST, P.
E16        2  AU=ERNEST, P. H.
E17        1  AU=ERNEST, P. H. H.
E18        2  AU=ERNEST, P. S.
E19        1  AU=ERNEST, P.-W.
E20        1  AU=ERNEST, P.H.H.
E21        1  AU=ERNEST, P.S.
E22        1  AU=ERNEST, PATRICIA
E23        1  AU=ERNEST, PATRICIA STROTHER
E24        1  AU=ERNEST, R. G.
```

inv

```
Ref    Items  Index-term
E1         0 *AU=FINDEN-BROWNE, C
E2         1  AU=FINDEN-BROWNE, C.
E3         1  AU=FINDEN, C.R.
E4         2  AU=FINDEN, P.
E5         1  AU=FINDEN, P. O.
E6         1  AU=FINDEN, R.E.
E7         3  AU=FINDEN, RANDY
E8         1  AU=FINDEN, WALTER FREDERICK
E9         7  AU=FINDENEGG, G.
E10        5  AU=FINDENEGG, G. H
E11       17  AU=FINDENEGG, G. H.
E12        1  AU=FINDENEGG, G. R.
```

inv

```
Ref    Items  Index-term
E1        14 *AU=MOORE, J
E2         2  AU=MOORE, J . DUNCAN, JR.
E3         1  AU=MOORE, J BERNARD
E4         1  AU=MOORE, J C
E5         1  AU=MOORE, J DANCAN JR
E6         3  AU=MOORE, J DUNCAN
E7       525  AU=MOORE, J DUNCAN JR
E8         1  AU=MOORE, J GEORGE
E9         1  AU=MOORE, J MICHAEL
E10        2  AU=MOORE, J PATRICK
E11        2  AU=MOORE, J S
E12        7  AU=MOORE, J STEVEN
```

Inv

```
Ref    Items  Index-term
E13        1  AU=MOORE, J. MICHAEL
E14        4  AU=MOORE, J. N
E15      194  AU=MOORE, J. N.
E16        2  AU=MOORE, J. O
E17        9  AU=MOORE, J. O.
E18        1  AU=MOORE, J. P
E19       15  AU=MOORE, J. P.
E20        2  AU=MOORE, J. R
E21       25  AU=MOORE, J. R.
E22        1  AU=MOORE, J. RICHARD
E23        3  AU=MOORE, J. ROBERT
E24        4  AU=MOORE, J. ROSS

         Enter P or PAGE for more
?p

Ref    Items  Index-term
E25        8  AU=MOORE, J. S
E26       56  AU=MOORE, J. S.
E27        1  AU=MOORE, J. SCOTT
E28        1  AU=MOORE, J. SHERROLD
E29        1  AU=MOORE, J. STEPHEN
E30        8  AU=MOORE, J. STEVEN
E31        3  AU=MOORE, J. STROTHER
E32        2  AU=MOORE, J. STUART
E33        1  AU=MOORE, J. T
E34       37  AU=MOORE, J. T.
E35        8  AU=MOORE, J. V.
E36      227  AU=MOORE, J. W.
```

inV

```
Ref    Items  Index-term
E13       84  AU=MOORE, J.M.
E14        3  AU=MOORE, J.M., JR.
E15        3  AU=MOORE, J.MCM.
E16       27  AU=MOORE, J.N.
E17       12  AU=MOORE, J.O.
E18       41  AU=MOORE, J.P.
E19       57  AU=MOORE, J.R.
E20        1  AU=MOORE, J.R. (EDITOR)
E21        4  AU=MOORE, J.R., JR.
E22       93  AU=MOORE, J.S.
E23        1  AU=MOORE, J.S., JR.
E24       46  AU=MOORE, J.T.
```

Inv

```
Ref    Items  Index-term
E1         2  AU=MOORE, JAKE
E2         1  AU=MOORE, JAME
E3       142  AU=MOORE, JAMES
E4         7  AU=MOORE, JAMES A
E5        11  AU=MOORE, JAMES A.
E6         1  AU=MOORE, JAMES A., JR.
E7         1  AU=MOORE, JAMES ALBERT
E8         1  AU=MOORE, JAMES ALEXANDER
E9         2  AU=MOORE, JAMES ALFRED
E10        1  AU=MOORE, JAMES ALLAN
E11        1  AU=MOORE, JAMES ALLEN
E12        1  AU=MOORE, JAMES ALTON
```

inv.

```
Ref    Items  Index-term
E1         6  AU=MOORE, JAMES P., JR.
E2         1  AU=MOORE, JAMES PAUL
E3         1  AU=MOORE, JAMES PETER
E4         1  AU=MOORE, JAMES PROCTOR
E5         1  AU=MOORE, JAMES R
E6         4  AU=MOORE, JAMES R.
E7         1  AU=MOORE, JAMES R., JR.
E8         2  AU=MOORE, JAMES RAY
E9         1  AU=MOORE, JAMES REESE
E10        1  AU=MOORE, JAMES RICHARD
E11        1  AU=MOORE, JAMES RUSSELL
E12        8  AU=MOORE, JAMES S
```

inv.

```
Set      Items     Description
S1           4     AU='ERNEST, M.'
S2           1     AU='FINDEN-BROWNE, C.'
S3          14     AU='MOORE, J'
S4          31     E20, E21, E22, E23
S5          62     E19, E20, E21                               Inv
S6         143     E2, E3
S7          11     E5, E6, E7, E8, E9, E10, E11
S8         261     S3 OR S4 OR S5 OR S6 OR S7
S9         266     S8 OR S1 OR S2
S10          1     S9 AND (INFORMATION? (1N) TECHNOLOG?)           term
S11    1240894     INFORMATION(1N)TECHNOLOG?
S12          1     S9 AND (IBM OR (INTERNATIONAL (1N) BUSINESS (1N) MACHINE?)) — inv/assignee
S13    1644801     IBM OR (INTERNATIONAL (1N) BUSINESS (1N) MACHINE?) — assignee
S14     130013     S13 AND S11
S15          3     S14 AND (WFMS)
S16       4738     S14 AND WORKFLOW?
S17        874     S14 AND ((USAGE OR TRANSACTION)(1N)DATA)
S18       3533     S16 NOT (PY>=2001 OR PD>=2001)
S19        172     S18 AND REUSE                                         term
S20        118     S19 AND SERVICE
S21          0     S18 AND (MONITOR? (1N) COMPONENT?)
S22         50     S18 AND (MONITOR? (4N) (COMPONENT? OR ACTIVIT?))
?
```

12/9,K/1 **(Item 1 from file: 15)**
DIALOG(R)File 15:ABI/Inform(R)

01378966 00-29953
Standardizing reuse
Rada, Roy; ****Moore, James****
Communications of the ACM v40n3 PP: 19-23 Mar 1997 ISSN: 0001-0782
JRNL CODE: ACM
DOC TYPE: Journal article LANGUAGE: English LENGTH: 5 Pages
WORD COUNT: 2395

ABSTRACT: The popular reuse icon (3 green arrows in a cycle) is typically about decomposing natural products and incorporating them in new natural products in a cyclical way. Software reuse, however, should lead to new products in a spiraling way. Reuse is the practice of using an asset in more than one software system. An asset is any product of the software life cycle. Reuse requires the existence of a library of assets. A reuse library is a controlled collection of assets, together with the procedures and support functions required to provide the assets for reuse. Reuse typically occurs within a domain of activity or knowledge in which applications share a set of common capabilities and data. Some companies keep their software reuse standards confidential. Government agencies have been active in advancing reuse standards that are shared with the public. The Department of Defense is already writing software development contracts that require reuse. Yet, no internationally recognized standard for how software reuse should occur exists.

TEXT: The distinction between use and reuse is sometimes a subtle one. We would argue that success in society is intimately linked, in the first instance, to the ability to create products and/or services that are used. The grander success occurs when what one produces becomes a critical building block in what others create-this is reuse.

The popular reuse icon (three green arrows in a cycle) is typically about decomposing natural products and incorporating them in new natural products in a cyclic way (see Figure 1). Software can, however, be arbitrarily often copied, and software reuse should lead to new products in a spiraling way (see Figure 2).

More rigorously, reuse is the practice of using an asset in more than one software system. An asset is any product of the software life cycle. Reuse requires the existence of a library of assets. A reuse library is a controlled collection of assets, together with the procedures and support functions required to provide the assets for reuse. Reuse typically occurs within a domain of activity or knowledge in which applications share a set of common capabilities and data.
While the terminology of assets, reuse libraries, and domains is germane to understanding the technological side of software reuse, software reuse is also about processes that involve people.

It is about learning how to achieve software reuse, about planning an organization's strategy for reuse, and about maintaining a process of reuse that people have been taught to follow. Reuse is also about economics.

Industry View

The increased size of the global market increases the potential for the number of units to be sold. This permits a business to justify increased capital investment in a product while lowering per-unit price, if penetration of a large percentage of the now larger potential market can be ensured.

However, penetration is crucially linked to being first to the market. Software developers, therefore, find themselves in a situation of coping with the commodity pricing of high-capitalization software with market share strongly linked to the speed of introduction. This trend is favorable to software reuse because reuse practices support the economical capitalization of development effort in a manner that can accelerate the introduction of new products to the market.

Experience in software development has frequently shown that the challenge to software reuse is less the development of new programming languages or technologies but rather the way an organization rewards software reuse on the part of its software engineers. Motorola software engineers in one division were given financial rewards for storing assets in a library and then given significant further rewards each time the asset was used in another product. Those incentives helped that division become the premier example of software reuse in Motorola.

(Illustration Omitted)

Projects at ****IBM**** have involved various sophisticated object-oriented approaches to creating assets, but one of the most potent conclusions of the ****IBM**** experiences was that tools and methods should integrate seamlessly into the work environment. The greatest impact on reuse came when ****IBM**** software engineers had the reuse library at their fingertips-they could easily move between relevant library contents and the problemsolving for which reuse was relevant.

Research by William Frakes ("Sixteen Questions about Software Reuse," Communications, June 1995) suggests that the mere presence of a usable library is "good enough" and that additional investment in cataloguing mechanisms and search tools provides little payback. This argues for purchasing a library tool or outsourcing a reuse library service and focusing one's own investment on the contents of the library rather than the nature of the tool.

There is another side to the industry reuse story. Customers may want to have access to reusable software libraries. In the World-Wide Web market, Microsoft, Netscape, and others are trying to gain market share with customers who want to gain access to reusable code libraries for building interactive Web sites. Having components in standardized languages that can interoperate in standard ways with various Web-related functionalities is important to such reuse libraries.

So some of these software houses are compelled to compete in being able to label their products as standard in a way that befits the generation of customer-accessible reuse libraries.

Government View

Companies often see their approach to internal software reuse standards as important to their competitive advantage. Some of these companies keep these software reuse standards confidential. Government agencies have played a particularly active role in advancing reuse standards that are shared with the public.

In the U.S., government's interest is particularly acute because of the nature of its business methods. By law, government procurements are required to be fair, a virtue valued beyond even effectiveness. In general, government cannot award a follow-on contract to a company simply because it performed well on the predecessor program-a fair competition among interested parties is required and the award made based on some combination of proposed actions and estimated cost rather than track record. In this sort of contracting environment, government has an essential need to ensure that one contractor can reuse the products of previous contracts.

The U.S. Department of Defense spends $30 billion a year on software. The military may buy software from many suppliers and yet wants what it buys from one supplier to be accessible for reuse by another supplier. The military has developed software reuse guidelines that can be used to standardize contractor performance. One salient project from the the Department of Defense Advanced Research Projects Agency is called Software for Adaptable, Reliable Systems (STARS).

One of the key documents from STARS is the Conceptual Framework for Reuse Processes (CFRP). CFRP defines a conceptual framework for reuse in terms of the processes involved. The framework is intended to be generic with respect to domains, organizations, economic sectors, methodologies, and technologies. CFRP identifies the processes involved in reuse and describes at a high level how those processes operate and interact. The management of reuse is described within a spiral of plan, enact, and learn (see Figure 3). For instance, in planning, one proposes measurements of library assets and their use. These measures then become integral to the evaluation of a reuse project and learning how to change the next plan.

Another STARS document is called "Reuse Strategy Model: Planning Aid for Reuse-Based Projects." The document is intended as a planning aid for reuse-based projects. It provides a set of dimensions for characterizing current reuse practice and a suggested process performing the characterization.

(Chart Omitted)

Captioned as: Figure I. Popular reuse icon

(Chart Omitted)
Captioned as: Figure 2. Spiraling reuse icon

Both of these STARS documents effectively describe principles, but implementation details are left to the organizations that want to adopt STARS. There is a huge gap between the general principles for how an organization might operate so as to achieve software reuse and the specific details of a company's implementation. This same phenomenon applies to ISO 9000 (see "ISO 9000 Reflects the Best in Standards," Communications, March 1996, p. 17), providing clear, compelling principles. However, implementation is a long distance from the very general principles of ISO 9000.

Various components of the U.S. military have developed documents to guide the implementation of software reuse within their area. For instance, in 1995 the Department of the Air Force issued "Guidelines for Successful Acquisition and Management of Software Intensive Systems." Software reuse is a central theme of the document and is explicitly discussed in several chapters.

Past NATO Efforts

In 1992 the North Atlantic Treaty Organization (NATO) issued three documents about software reuse as standards for NATO:

"Standard for the Development of Reusable Software Components" provides prescriptive guidance for structuring a software development process that leads to the development of reusable software assets. The document addresses requirements analysis, design principles, detailed design and implementation, quality assurance and test, and documentation. An appendix provides detailed reusability guidelines for the Ada programming language.

"Standard for Management of a Reusable Software Component Library" provides guidance for the establishment and operation of a software reuse support organization. The document provides management guidance for the operation of a software reuse support organization including the activities of requirements analysis, asset accession, configuration management, the management of automated library tools, and the management of the library organization.

"Standard for Software Reuse Procedures" provides guidance for reuse of NATO reuse library assets.

When one reads the NATO documents, one appreciates that they are NATO-specific. In other words, they are plans for how NATO itself will conduct software reuse activities.

The observation about the NATO-specific standards is merely one example of a more general phenomenon in reuse standardization-the most tangible reuse standards are specific to a single organization. That's because successful reuse practices touch many parts of an organization's methods and culture for doing business.

To better appreciate the organization-specific character of the most tangible reuse guidelines, consider an example. We might write a standard saying that any reusable subroutine must be documented with its "intended function." In an organization indoctrinated in Harlan Mills's structured programming techniques, this has a precise mathematical definition that is well understood because of the corporate culture of pursuing Mills's techniques. If we were to generalize this standard to other organizations and expect them to use Mills's techniques, we would be faced with the complaint that "this organization documents its software differently." So in an effort to be general we would relax the standard to a requirement that one write down what the software is supposed to do-a very nonspecific requirement that can be satisfied in a variety of superficial ways. In broadening the applicability of the standard, we have robbed it of its ability to discriminate. We see this phenomenon over and over again in software reuse standards.

(Chart Omitted)

Captioned as: Figure 3.

Upcoming Standards

Standards can be developed by many different kinds of organizations (see "Standards: Free or Sold" Communications, Feb. 1995, p. 23). ****IBM**** and Motorola have developed corporate standards for reuse among their employees. Government agencies have developed standards for how government contractors must follow software reuse processes. Relatively little has been done explicitly about software reuse by official standards development organizations.

The major international standards organization relative to information technology is ISOIEC JTC1. The standards from the U.S. Military and NATO are not JTC 1 standards. JTC 1 has one project under way that is particularly germane to software reuse. This is the Software Process Improvement Capability dEtermination (SPICE) project.
SPICE provides road maps for important software practices. Reuse is one of the practices covered by SPICE. SPICE is designed to provide software organizations with an internationally recognized mechanism to support their continuous process improvement programs, and to help managers ensure that the process is aligned with the business needs of the organization. SPICE will also help purchasers determine the capability of software suppliers and identify risks. (Bell Canada has a corporate process maturity assessment standard that includes software reuse and is an important reference for SPICE.)

One of the major contributors of software standards is the IEEE Software Engineering Standards Committee (SESC). Most of the products of the SESC that are related to reuse are submitted by the Reuse Library Interoperability Group (RIG). The RIG is an independent group of reuse library users, operators, and vendors that are developing specifications for the interoperation of reuse libraries. The RIG is comprised of more than 150 individual members and about 20 organizational members participating at various levels of activity. The RIG's work is progressed to the status of standards via its Memorandum of Understanding with the IEEE SESC.

The RIG has produced a data model for reuse library interoperability-Basic Interoperability Data Model (BIDM) to describe the minimal information that reuse libraries should be prepared to exchange in describing their assets.This document is already an IEEE Standard. The RIG has also developed extensions to the BIDM that may be used for explaining the manner in which assets were certified for quality or other attributes and guidance for performing certification.

The RIG's products concern only the ability for networks of reuse libraries to share assets. To chart its course for a broader scope of reuse standards, the IEEE SESC chartered a Reuse planning group to develop a plan. That plan recommends four new efforts:

1. A standard describing principles of software reuse-principles that individual organizations could satisfy in a variety of ways.

2. A standard describing the characteristics of domain analysis-domain analysis though little understood is now frequently practiced.

3. A supplement to the ISO 12207 life cycle process standard (see "Organizational Badge Collecting," Communications, Aug. 1996, p. 17) to describe how reuse practices relate to the 12207 processes.
4. A method for performing capability assessment for suppliers of reusable components, probably within the context of

SPICE.

These recommendations are notable also for the efforts that were considered but rejected. For example, the group rejected efforts for standards related to "reuse development practices" because they were regarded as being too specific to tools and technologies and being too lowlevel and voluminous. Standards for library organization and operation were rejected because reliable measures of effectiveness do not yet exist. Guides for reuse adoption were rejected because there is little evidence of repeatable effectiveness and because there is little evidence of de facto consensus.

Perhaps most notably, an effort to write a standard for the assessment of reusable components was rejected because of the concerns cited earlier in this column regarding the ability to "scale" reuse practices beyond the scope of a single organization.

Conclusion

The U.S. Department of Defense is already writing software development contracts that require reuse. Yet, no internationally recognized standard for how software reuse should occur exists. Such standards are needed.

Two kinds of standards are being developed. One kind is technical and focuses on the software assets that are to be reused. Another kind is social and guides the human side of software reuse. The artifacts are much less complex than the people. Developing a suite of standards that apply seamlessly to the people and their artifacts will not be easy. El

ACKNOWLEDGEMENTS

This column has borrowed heavily from the Action Plan: Reuse Planning Group of the SESC of the IEEE Computer Society. Citations not present there can be found in the book Software Reuse (Roy Rada, Intellect Books, 1995).

Author Affiliation:

ROY RADA is Boeing Distinguished Professor of Software Engineering at Washington State University. JAMES MooRE is Special Assistant for the Application of Software Standards and Reuse at the Mitre Corp

GEOGRAPHIC NAMES: US

DESCRIPTORS: Standardization; Software; Information management; Product development; International
CLASSIFICATION CODES: 5320 (CN=Quality control); 9180 (CN=International); 5240 (CN=Software & systems); 9190 (CN=United States); 8302 (CN=Software and computer services); 7500 (CN=Product planning & development)

...****Moore, James****
...TEXT: that division become the premier example of software reuse in Motorola.

(Illustration Omitted)

Projects at ****IBM**** have involved various sophisticated object-oriented approaches to creating assets, but one of the most potent conclusions of the ****IBM**** experiences was that tools and methods should integrate seamlessly into the work environment. The greatest impact on reuse came when ****IBM**** software engineers had the reuse library at their fingertips-they could easily move between relevant...many different kinds of organizations (see "Standards: Free or Sold" Communications, Feb. 1995, p. 23). ****IBM**** and Motorola have developed corporate standards for reuse among their employees. Government agencies have developed...